

# Grupisanje više elemenata u jedan block element (DIV)

`<div>` element nam omogućuje da grupišemo više elemenata u jedan block element. Na primer, možemo kreirati *div* za header strane, i sve header elemente (logo, slogan, navigacija, pretraga) postaviti unutar njega.

`<div> ... </div>`

# Grupisanje više elemenata u jedan block element (DIV)

- Div tag, pošto je block tip elementa, počinje na novom redu.
- Po default vrednostima div-ovi nemaju nikakvu stilizaciju (transparentna je pozadina, border, margine i padding su 0, visina zavisi od sadržaja, širina je maksimalna).
- Ipak, možemo im dodati ID i/ili CLASS vrednosti i preko njih ih „gađati“ CSS opisima. Ta jednostavnost u osnovi, ali sa druge strane, velike mogućnosti stilizacije sa druge, čine da uz novije semantičke elemente **div elementi budu osnova građe svakog sajta.**
- Div elemenat (boks) može sadržati i više drugih elemenata, kreirajući hijerarhiju, što predstavlja ugnježdene elemente stranice. Prozor čitača u ovoj hijerarhiji predstavlja Root (koreni) element.

# Grupisanje više elemenata u jedan inline element (SPAN)

`<span>` element je inline ekvivalent `<div>` elementa. Pomoću njega možemo grupisati više inline elemenata u jedan inline element. Koristi se najčešće za izdvajanje delova teksta, jer možemo, na primer, označiti deo teksta, dodati CLASS ili ID i zatim CSS opisom uticati na taj deo teksta (koji je okružen span-om).

`<span> ... </span>`

# Grupisanje više elemenata u jedan inline element (SPAN)

- Ono što je specifično za span je da **ne sme** sadržati block elemente, već samo druge inline elemente.
- U w3 specifikaciji za HTML se kaže:  
*“Generally, block-level elements may contain inline elements and other block-level elements. Generally, inline elements may contain only data and other inline elements. Inherent in this structural distinction is the idea that block elements create "larger" structures than inline elements”.*
- Dakle, **block elementi mogu sadržati inline i block elemente, dok inline elementi mogu sadržati samo druge inline elemente.**

# CSS Box model

- Kao što je već pomenuto, oko svakog block elementa na stranici, kreira se imaginarni okvir (box) koji možemo stilizovati.
- Ako bi razložili na korake prikaz nekog elementa, dobili bi ovakav redosled:
  1. Html postavlja sadržaj i kreira imaginarni okvir.
  2. CSS taj okvir stilizuje, menja po potrebi.
  3. Taj element se prikazuje korisniku.

# CSS Box model (samo za block elemente!)

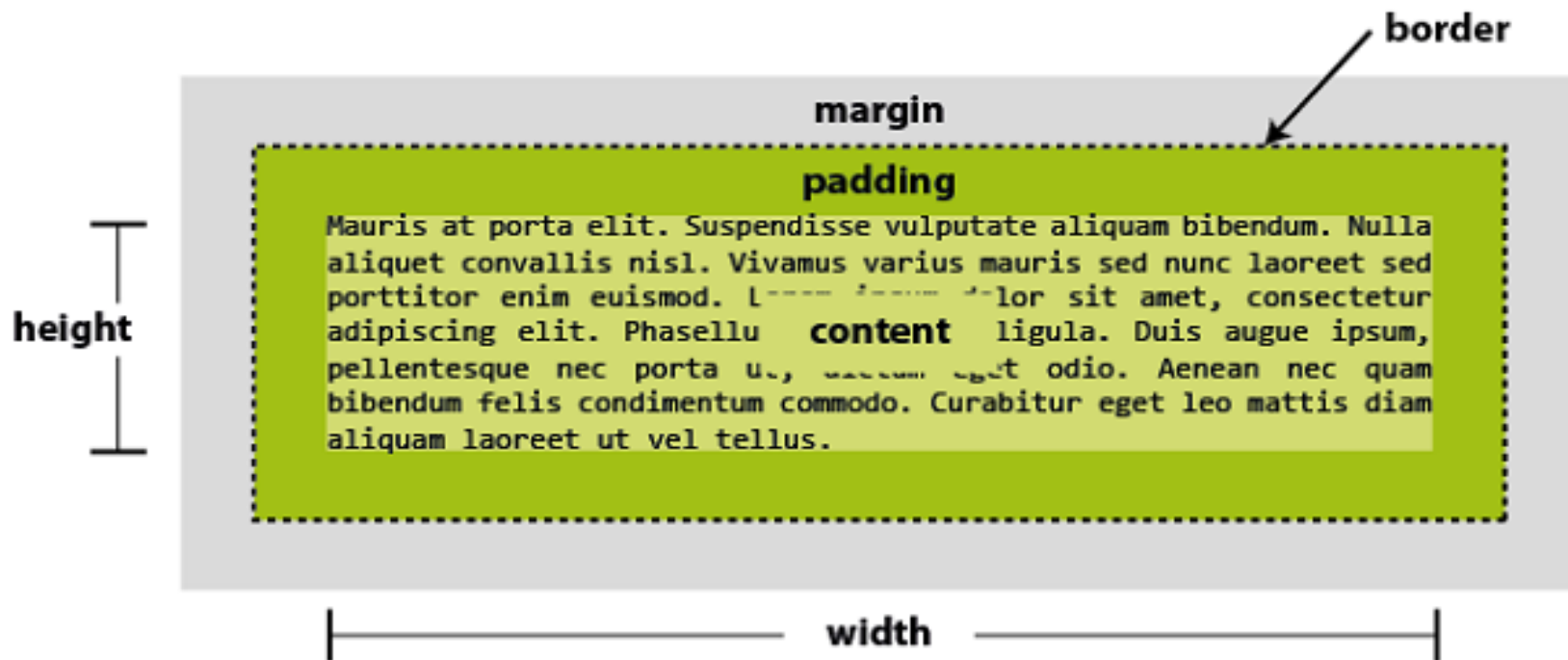
Box Model se sastoji od pet osnovnih svojstava elemenata:

- **Width** (širina)
  - **Height** (visina)
  - **Margin** (margine ili spoljni razmak)
  - **Padding** (padding ili unutrašnji razmak)
  - **Border** (okvir, ivica)
- 
- Iako ne spada u CSS Box model, često se pridodaje i **Background** svojstvo (pozadina).

Pogledajmo na primeru. Ovo je jedan stilizovan paragraf teksta:

Mauris at porta elit. Suspendisse vulputate aliquam bibendum. Nulla aliquet convallis nisl. Vivamus varius mauris sed nunc laoreet sed porttitor enim euismod. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at sodales ligula. Duis augue ipsum, pellentesque nec porta ut, dictum eget odio. Aenean nec quam bibendum felis condimentum commodo. Curabitur eget leo mattis diam aliquam laoreet ut vel tellus.

I on sadrži sve elemente CSS box modela, a koji su označeni na sledećoj slici:





# Delovi Box modela

**Margine** su uvek transparentne i odvajaju element od drugih elemenata na stranici. Okviri (**Borders**) mogu biti različitih stilova i definišu granice elementa. **Padding** odvajaja sadržaj elementa (Content) od njegovih okvira. Background podešavanja elementa se odnose na oblast unutar okvira zajedno sa **Padding i Content** delom (na slici Padding deo je prikazan u nešto tamnijoj zelenoj boji, mada je realno nevidljiv).

# Delovi Box modela

- Margins, Borders i Padding svojstva za svaku stranu elementa (Top, Right, Bottom, Left) mogu biti različite vrednosti.
- Po default vrednostima, ova svojstva su u većini slučajeva 0.
- Margine mogu imati i negativnu vrednost, mada to nikako nije preporučeno.

# CSS Box model „matematika“

- Važno je zapamtiti da Width i Height vrednosti određuju samo širinu, odnosno visinu sadržaja (content) elementa.
- **Realna visina i širina** koje elemenat zauzima na strani je **Content + Padding + Border**.
- Na primer, ukoliko imamo div element, u kome je sadržaj 100 x 100 piksela, padding je 10px (sa svih strana), a border 2px (takođe sa svih strana), mesto koje takav elemenat zauzima je 124 x 124 piksela. To je zato što je širina ukupno: **100px** za sadržaj + **10px za padding x2** zbog levog i desnog paddinga + **2px za border x2** jer imamo levu i desnu. ( $100 + 10 \times 2 + 2 \times 2 = 124\text{px}$ ). Isto je i za visinu.

# CSS Box model „matematika“

CSS kôd za element iz ovog primera bi mogao biti:

```
.mojDiv {  
    width: 100px;  
    height: 100px;  
    padding: 10px;  
    margin: 0;  
    border: 2px solid #f90;  
}
```

# width i height

- Ova dva svojstva predstavljaju širinu, odnosno visinu sadržaja elementa. Po osnovnim vrednostima, širina i visina su taman tolike da uokvire sadržaj (inline), odnosno širina maksimalna, a visina spram sadržaja (block).
- Vrednost može biti u **pikselima, procentima, em vrednostima i td.** Pikseli su najpopularniji način i u većini slučajeva se oni koriste. Ukoliko koristimo procenete, veličina je određena veličinom prozora browsera, odnosno veličinom roditeljskog elementa (ako postoji). Kod em vrednosti, veličina zavisi od veličine teksta (fonta) unutar njega.

```
.mojDiv {  
    width:400px;  
    height:80%;  
}
```

# Padding

Ovo svojstvo definiše unutrašnji razmak (padding). Ukoliko ne postavimo vrednost, podrazumeva se 0. Ako svojstvo ne sadrži sufiks, onda se odnosi na sve četiri strane:

**padding:10px;**

Ukoliko želimo različite padding vrednosti gore, desno, levo i dole možemo pisati na primer:

**padding-top:10px;**

**padding-right:15px;**

**padding-bottom:20px;**

**padding-left:25px;**

Postoji i skraćeni način pisanja (shorthand) pomoću kojeg u jednom redu, koristeći samo padding svojstvo (bez sufiksa) možemo odrediti sve četiri vrednosti. Kod shorthand padding svojstva možemo pisati:

**padding:10px 15px 20px 25px;**

Ovako napisana deklaracija će imati isti rezultat kao malopredašnje četiri. Važno je samo naglasiti da se vrednosti **uvek** pišu u pravcu kazaljke na satu počevši od gornje. U našem primeru, 10px je vrednost gornjeg, 15px desnog, 20px donjeg, a 25px levog paddinga.

# Margin

Ovo svojstvo definiše spoljašnji razmak (margin). Sva pravila koja važe za padding, važe i za margin. Podrazumevana vrednost je 0. Možemo pisati opšte svojstvo (margin), ili koristiti iste sufikse za strane kao kod paddinga. Ili pak možemo pisati shorthand (skraćeno) svojstvo.

```
...  
margin:100px;  
/* sve iste */  
...
```

```
...  
margin-top:100px;  
margin-right:30px;  
margin-bottom:200px;  
margin-left:25px;  
...
```

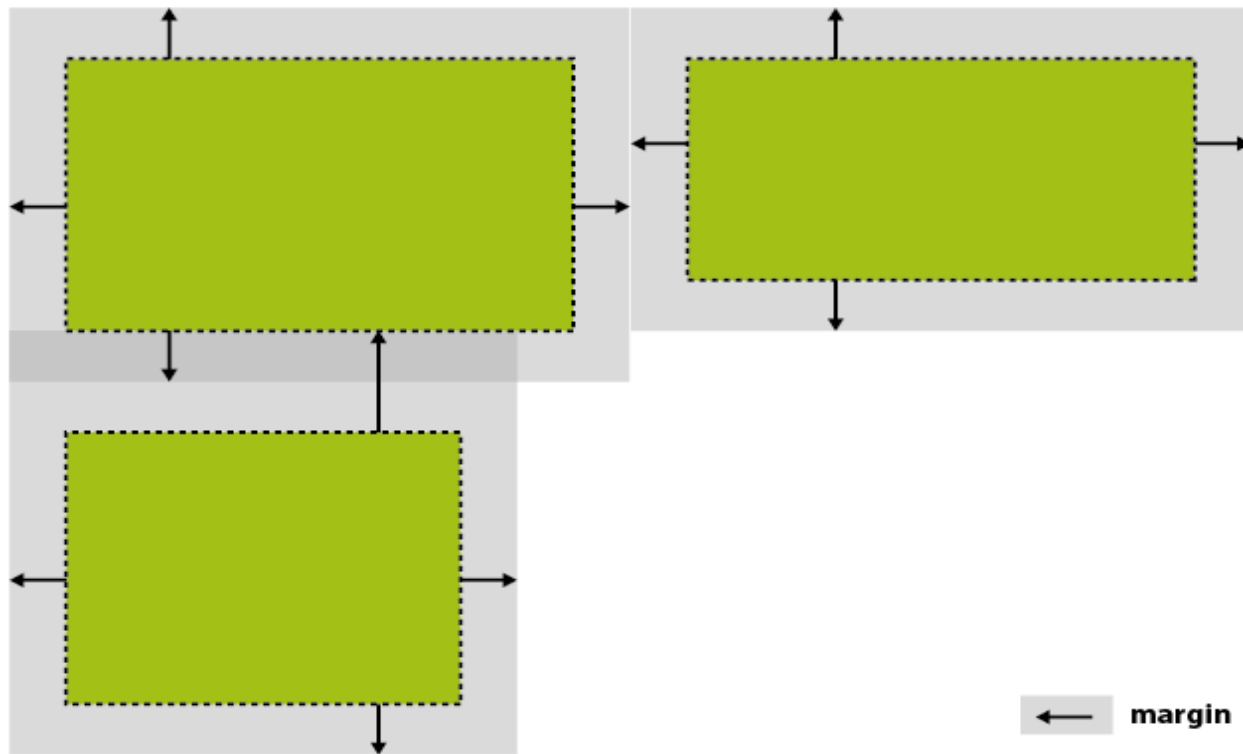
```
...  
margin:100px 30px 200px 25px;  
...
```

# Margine i collapse

Kod margina je bitno znati da one mogu **kolapsirati (collapse)**, odnosno, ukoliko se nađu dva elementa jedan ispod drugog, ukoliko postoje margine između njih (donja margina na gornjem i gornja margina na donjem elementu), ukupan prostor **neće** biti zbir njihovih margina, već će se uzeti vrednost veće od te dve. Nasuprot tome, ukoliko su elementi jedan pored drugog, margine ne kolapsiraju, već se sabiraju.



# Margine i collapse



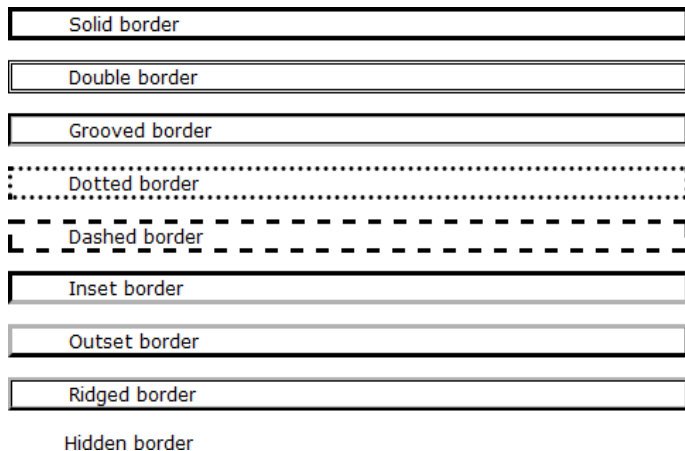
# Border-width

Ovo svojstvo koristimo kako bi definisali debljinu ivice oko elementa. Kao vrednost možemo koristiti piksele ili jednu od tri vrednosti: *thin*, *medium*, *thick*. Procenti i ostale jedinice u ovom slučaju nisu dozvoljene. Poput margine i paddinga, i kod border svojstva vrednost se odnosi na sve četiri strane. Različite možemo pisati skraćeno (shorthand) ili odvojeno, na primer:

```
border-top-width: 6px;  
border-right-width: medium;  
border-bottom-width: 3px;  
border-left-width: thin;
```

# Border-style

Ovim svojstvom određujemo tip ivice, okvira (border) oko elementa. Na slici ispod vidimo moguće varijante. Uglavnom se koristi *solid* (obična linija). Vrednost je sam naziv tipa (solid, double, grooved i tako dalje).



# Border-style

- Ukoliko su sve ivice elementa iste, onda pišemo, na primer  
**border-style:solid;**
- Takođe, možemo individualno odrediti tip ivice koristeći:  
**border-top-style: solid;**  
**border-right-style: groove;**  
**border-bottom-style: dotted;**  
**border-left-style: outset;**

# Border-color

Kao što samo ime kaže, ovo svojstvo definiše boju ivice. Oznaku boje (vrednost) možemo pisati na različite načine, pomoću hex, rgb sistema ili imena boje.

**border-color: #FF9900;**

# Border

- Možemo iskombinovati i skratiti sve border varijante.
- Na primer, možemo pisati:

**border: 1px solid #FF9900;**

# Display

Kao što znamo, block elemente možemo pretvarati u inline i obrnuto.

Tip elementa može biti određen korišćenjem svojstva **Display** kroz CSS. Ako na primer, svojstvu Display nekog Inline elementa dodelimo vrednost Block, tada će taj element biti tretiran kao i svaki drugi Blok element.

```
span {  
    display:block;  
}
```

Ovim primerom smo span elemente pretvorili u block.

# Visibility

Visibility svojstvo nam omogućuje da sakrijemo određeni element, ali da pritom ostane rezervisan prostor za njega. Može imati jednu od dve vrednosti, *hidden* (sakriva element) ili *visible* (prikazuje element – ovo je default vrednost i ne moramo je pisati).

```
span {  
    visibility:hidden;  
}
```

Razlika između *visibility:visible;* i *display:none;* je u tome što u prvom slučaju element nestaje, ali ostaje rezervisan prostor za njega, i ostali elementi se ne premeštaju, dok u drugom slučaju, element nestaje sa strane, ali i prostor rezervisan za njega, te se i ostali elementi razmeštaju po strani.



# Overflow

Ovim svojstvom definišemo šta će se desiti ukoliko sadržaj prelazi okvire koji su mu zadati. Na primer, možemo definisati širinu i visinu nekog div-a na 200x300px, a pritom uneti celu stranu teksta. Pošto su definisane dimenzije manje od potrebnih, deo teksta će biti van okvira. Koristeći overflow svojstvo možemo definisati da se sadržaj koji ide van elementa ne prikazuje pomoću *overflow:hidden;* ili da skroluje pomoću *overflow:scroll;* Još jedna moguća vrednost je *overflow:visible;* koja je primenjena po defaultu.

```
div.mojTekst {  
  overflow:scroll;  
}
```